

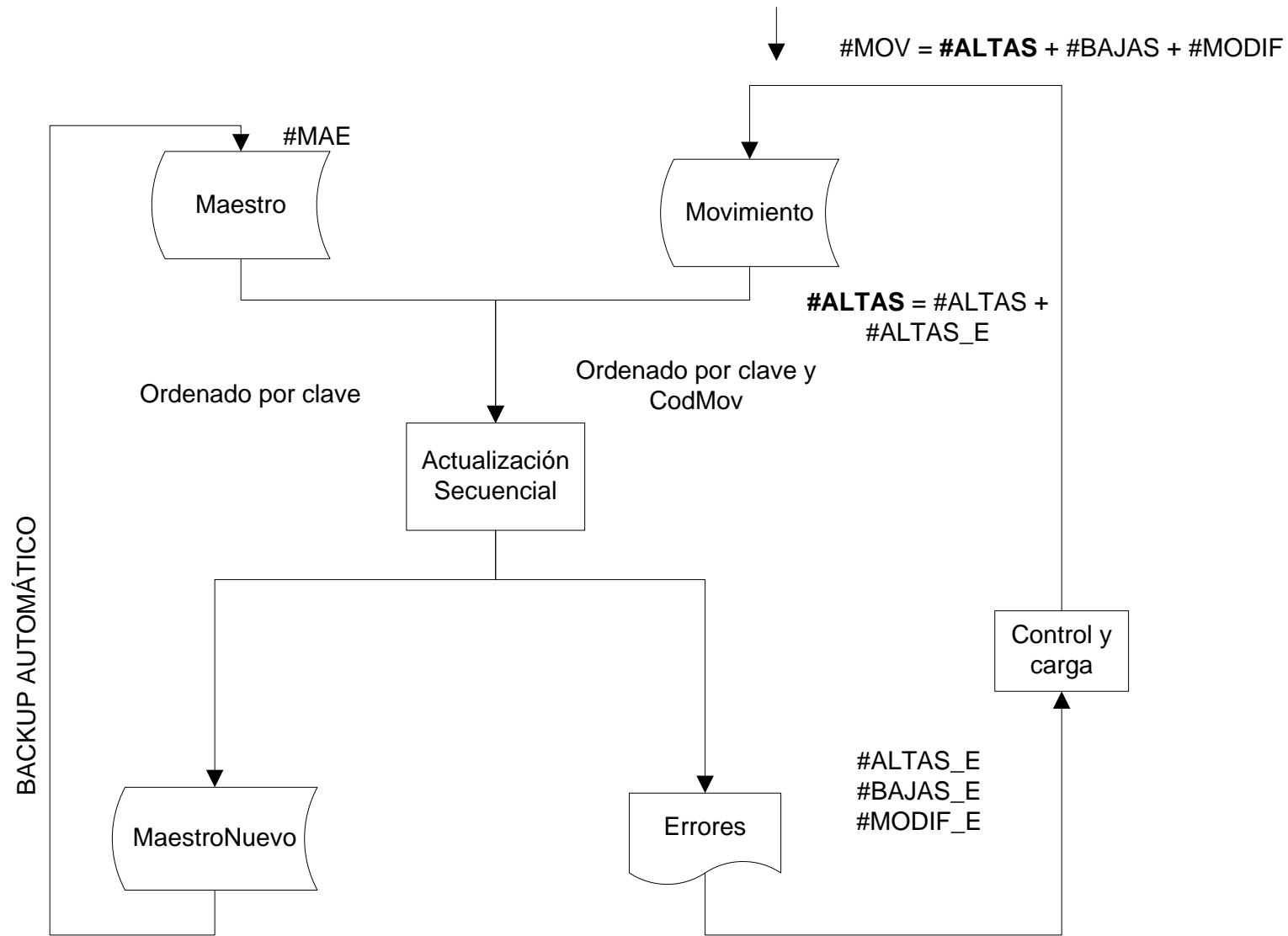


PROCESOS MÚLTIPLES CON ARCHIVOS

Actualización

UNIVERSIDAD TECNOLÓGICA NACIONAL
Ingeniería en Sistemas de Información
Algoritmos y Estructuras de Datos

ESQUEMA ACTUALIZACIÓN SECUENCIAL



EJEMPLO ACTUALIZACION SECUENCIAL UNITARIA

MAESTRO

ID CLIENTE	NOMBRE	DIRECCIÓN	LIMITE CREDITO
1236	SUSANA GIMENEZ	MONTEAGUDO 456	12000
5631	MIRTHA LEGRAND	LINIERS 2345	235
145233	WILLY SHAKI	VEDIA 123	3256

MOVIMIENTO

ID CLIENTE	LIMITE CREDITO
5631	5632
145233	1200

MAESTRO ACTUALIZADO

ID CLIENTE	NOMBRE	DIRECCIÓN	LIMITE CREDITO
1236	SUSANA GIMENEZ	MONTEAGUDO 456	12000
5631	MIRTHA LEGRAND	LINIERS 2345	5632
145233	WILLY SHAKI	VEDIA 123	1200

ACTUALIZACIÓN UNITARIA

Acción **ACTUNIT** Es
Ambiente

formato1= registro

campo1:....

campo2:....

campo3:.....

....

campoN:....

Marca_baja:.....

freg;

formato2= registro

campo1:

cod_mov: (A,B,M)

campo2:....

....

campoN:....

freg;

Amae: Archivo de formato1 ordenado por campo1;

Amov: Archivo de formato2 ordenado por campo1 y cod_mov;

Reg_mae, Reg_sal: formato1;

Regmov: formato2;

Subacción Leer_Movimiento es

LEER(Amov, Reg_mov)

SI FDA(Amov) ENTONCES

campo1:=HV

FSI;

FAcción.

Subacción Leer_Maestro es

LEER(Amae, Reg_mae)

SI FDA(Amae) ENTONCES

campo1:=HV

FSI;

FAcción.

ACTUALIZACIÓN UNITARIA

Algoritmo

Abrir_Archivos

Leer_Maestro

Leer_Movimiento

MIENTRAS (Reg_mae.campo1 <> HV) o (Reg_mov.campo1 <> HV) **HACER**

SI Reg_mae.campo1 = Reg_mov.campo1 **ENTONCES**

 Proceso_iguales

CONTRARIO

SI Reg_mae.campo1 < Reg_mov.campo1 **ENTONCES**

 Reg_sal := Reg_mae

 ESCRIBIR(Arch_sal, Reg_sal)

 Leer_Maestro

CONTRARIO

 Proceso_distintos

Fin_si;

Fin_Si;

Fin_mientras;

Cerrar Archivos

FinACCIÓN.



Un
movimiento
por cada
registro del
maestro

SUBACCIONES >> ACTUALIZACIÓN

Acción **Proceso_iguales** es

SI Cod_Mov = 'ALTA' ENTONCES

 ESCRIBIR('Error alta no existe')

 Reg_sal := Reg_mae

 ESCRIBIR(Arch_sal, Reg_sal)

CONTRARIO

 SI Cod_Mov = 'MODIF' ENTONCES

 Proceso_modif_maestro

 Reg_sal := Reg_mae

 ESCRIBIR(Arch_sal, Reg_sal)

 CONTRARIO * eliminación lógica *

 Marcar_registro

 Reg_sal := Reg_mae

 ESCRIBIR(Arch_sal, Reg_sal)

Fsi;

Fsi;

Leer_Maestro

Leer_Movimiento

FAcción.

Acción **Proceso_distintos** es

SI Cod_Mov = 'BAJA' ENTONCES

 ESCRIBIR('Error baja no existe')

CONTRARIO

 SI Cod_Mov = 'MODIF' ENTONCES

 ESCRIBIR('Error modificación no existe')

CONTRARIO

 Reg_sal.clave := Reg_mov.clave

 Reg_sal.campo1 := Reg_mov.campo1

 Reg_sal.campo2 := Reg_mov.campo2

 Reg_sal.campoN := Reg_mov.campoN

 Reg_sal.Marca_baja := ''

 ESCRIBIR(Arch_sal, Reg_sal)

Fsi;

Fsi;

Leer_Movimiento

FAcción.

SUBACCIONES >> ACTUALIZACIÓN

Subacción **Proceso_modif_maestro** es

Si Regmov.campo1 <> ' ' ENTONCES

Reg_mae.campo1 := Regmov.campo1

fsi;

Si Regmov.campo2 <> ' ' ENTONCES

Reg_mae.campo2 := Regmov.campo2

fsi;

Si Regmov.campo1 <> ' ' ENTONCES

Reg_mae.campo3 := Regmov.campo3

fsi;

* ... y así sucesivamente para todos los campos del registro...*

FinSubaccion.

Subacción **Marcar_registro** es

Reg_mae.Marca_baja:= '*'

* en vez de asterisco, se puede asignar la fecha del día, o cualquier otro dato, según el problema *

FinSubacción.

EJEMPLO ACTUALIZACION SECUENCIAL POR LOTES

MAESTRO

ID CLIENTE	NOMBRE	DIRECCIÓN	LIMITE CREDITO
1236	SUSANA GIMENEZ	MONTEAGUDO 456	12000
5631	MIRTHA LEGRAND	LINIERS 2345	235
145233	WILLY SHAKI	VEDIA 123	3256

MOVIMIENTO

ID CLIENTE	LIMITE CREDITO
5631	5632
5631	5800
5631	6200
145233	1200
145233	200

MAESTRO ACTUALIZADO

ID CLIENTE	NOMBRE	DIRECCIÓN	LIMITE CREDITO
1236	SUSANA GIMENEZ	MONTEAGUDO 456	12000
5631	MIRTHA LEGRAND	LINIERS 2345	6200
145233	WILLY SHAKI	VEDIA 123	200

ACTUALIZACIÓN POR LOTES

Algoritmo

Abrir_Archivos
Leer_Maestro
Leer_Movimiento

MIENTRAS (Clave_Mae <> HV) o (Clave_Mov <> HV) HACER
SI Clave_Mae < Clave_Mov ENTONCES * Mae sin Movimiento *

Reg_sal := Reg_mae
ESCRIBIR(Arch_sal, Reg_sal)
Leer_Maestro

CONTRARIO

SI Clave_Mae = Clave_Mov ENTONCES
Aux := Reg_mae * Mae con Movimiento *
MIENTRAS Clave_Mae = Clave_Mov HACER
Proceso_Movim
Leer_Movimiento
FMientras
Reg_sal := Aux
ESCRIBIR(Arch_sal, Reg_sal)
Leer_Maestro

CONTRARIO

Aux.clave := Reg_mov.clave
Aux.campo1 := Reg_mov.campo1
Aux.campo2 := Reg_mov.campo2

.....

Aux.campoN := Reg_mov.campoN
Aux.Marca_baja := ' '

Leer_Movimiento

MIENTRAS Clave_Aux = Clave_Mov HACER
Proceso_Movim
Leer_Movimiento

FMientras

Reg_sal := Aux
ESCRIBIR(Arch_sal, Reg_sal)

FSI;

FSI;
F Mientras;
Cerrar(Arch_sal)
FACCIÓN

ACTUALIZACIÓN POR LOTES

Subacción **Leer_Movimiento** es

LEER(Arch_mov, Reg_mov)

SI FDA(Arch_mov) ENTONCES

Clave_mov := High_value

FSI;

FAcción.

Subacción **Leer_Maestro** es

LEER(Arch_mae, Reg_mae)

SI FDA(Arch_mae) ENTONCES

Clave_mae := High_value

FSI;

FAcción.

Subacción **Proceso_Movim** es

SI Cod_Mov = 'M' ENTONCES

Proceso_modif_maestro

CONTRARIO

SI Cod_Mov = 'B' ENTONCES

Marcar_registro

Fsi;

FSI;

FAcción.

Subacción **Marcar_registro** es

Aux.Marca_baja:= '*'

FAcción.

ACTUALIZACIÓN POR LOTES

Acción **Proceso_modif_maestro** es

Si Reg_Mov.campo1 <> ' ' ENTONCES
Aux.campo1 := Reg_mov.campo1

fsi;

Si Reg_Mov.campo2 <> ' ' ENTONCES
Aux.campo2 := Reg_mov.campo2

fsi;

Si Reg_Mov.campo1 <> ' ' ENTONCES
Aux.campo3 := Reg_mov.campo3

fsi;

* ... y así sucesivamente para todos los campos del registro...*

FAcción.